

MEĐIMURSKO VELEUČILIŠTE U ČAKOVCU  
PROGRAMSKO INŽENJERSTVO

DRAŽEN VUK

IZRADA UPRAVLJAČKOG I SIGURNOSNOG WEB SUSTAVA ZA  
NADZOR PROSTORIJA UPOTREBOM MIKROKONTROLERA

ZAVRŠNI RAD

Čakovec, 2015.

MEĐIMURSKO VELEUČILIŠTE U ČAKOVCU  
PROGRAMSKO INŽENJERSTVO

DRAŽEN VUK

WEB CONTROL AND SECURITY SYSTEM FOR HOME  
SUPERVISING PURPOSE USING MICROCONTROLLERS

ZAVRŠNI RAD

Mentor:

mr. sc. Sanja Brekalo, v. predavač

Čakovec, 2015.

## Sažetak

*U ovom će završnom radu biti riječi o komunikaciji između korisnika i sustavu za nadzor prostorija uz pomoć raznih mikrokontrolera, Android telefona i internet servera. U eksperimentalnom je dijelu rada izrađen sustav koji služi za zaštitu prostora od poplave, provale, požara, komunikaciju s plinskim ili strujnim bojlerom pri podešavanju sobne temperature, implementaciju bežičnih senzora pokreta i Android aplikacije koja komunicira preko interneta s mikrokontrolerima. Na taj način korisnika upozorava u svakom trenutku o stanjima senzora, kao što je temperatura u stanu, je li plinsko kuhalo isključeno ili nije, o mogućnosti požara, poplave, provale. U slučaju da senzor detektira opasnost od bilo koje od navedenih prijetnji, sustav automatski obavještava korisnika putem mobilnoga uređaja o kojoj se nepogodnosti radi.*

*Za izradu sustava za nadzor prostorija, korišten je Arduino uređaj povezan s raznim senzorima. Dobivene vrijednosti senzora šalju se na udaljeni web server pomoću Arduino Ethernet Shield uređaj i interneta. Za pohranu podataka o stanjima senzora korištena je MySql baza podataka koja uz pomoć PHP skripte i (engl. Web Request) metode komunicira s Arduino uređajem.*

*Korisničko je sučelje realizirano pomoću HTML i PHP tehnologije, a prilagođeno je računalima i mobilnim uređajima. Android aplikacija također koristi takozvani (engl. Web View) za prikaz korisničkog sučelja, ali njezina je glavna zadaća komunikacija u određenim intervalima s bazom podataka i prepoznavanje prijetnje u objektu. Nakon prepoznavanja prijetnje u objektu, pozadinski proces Android aplikacije aktivira zvono, vibraciju telefona i porukom ukazuje o kakvoj se nepogodnosti radi.*

*Također, razrađen je temelj u komunikacijskom kanalu i u bazi podataka za moguće dodatne module koji se mogu jednostavno povezati, kao što su detektor zvuka te razni prekidači kojima se može upravljati preko internet preglednika ili Android aplikacije.*

*Ključne riječi: Arduino, mikrokontroler, Android, senzor, PHP, Java*

## Sadržaj

1. Uvod .....	1
2. Cilj rada istraživanja.....	2
3. Opis korištenih tehnologija.....	3
3.1. Mikrokontroler.....	3
3.2. Arduino .....	3
3.2.1. Osnovne funkcije Arduina setup() i loop() .....	3
3.2.2. Varijable Arduino sustava.....	5
3.3. Android .....	6
3.3.1. Android SDK.....	7
3.4. PHP .....	9
3.4.1. Prednosti PHP jezika.....	9
3.5. MySql baza podataka.....	10
3.5.1. Prednosti MySQL-a.....	10
3.5.2. Nedostaci MySQL baza podataka .....	11
4. Potrebni uređaji za realizaciju projekta .....	12
4.1. Arduino .....	12
4.2. Arduino Ethernet Shield .....	14
4.3. Senzor za temperaturu DHT11 .....	17
4.4. Senzor za vodu HV3N (Funduino) .....	17
4.5. Releji ulaz 5V .....	18
4.6. Bežičan wireless senzor pokreta .....	18
4.7. Senzor vatre .....	18
5. Praktični rad.....	19
5.1. Arduino aplikacija.....	19
5.1.1. Funkcija setup .....	19

5.1.2. Funkcija loop.....	20
5.1.3. Funkcija WebRequest .....	20
5.1.4. Funkcija PaliGasiAlarm .....	22
5.1.5. Provjera poplave i analogno učitavanje podataka .....	22
5.1.6. Provjera pokreta .....	23
5.2. Android aplikacija.....	24
5.3. Internet aplikacija .....	27
6. Cijena sustava.....	30
7. Zaključak .....	31
8. Literatura .....	32

## 1. Uvod

Uz korištenje pametnih telefona i ostalih suvremenih tehnologija, ljudi se počinju navikavati na sve što im skraćuje vrijeme obavljanja svakodnevnih zadataka, smanjuje troškove i čini im život sigurnijim i lagodnijim. Današnje je društvo orijentirano na stvaranje vrijednosti i svako poboljšanje tehnologija daje pozitivne rezultate u sigurnosnom i ekonomskom smislu što neminovno vodi poboljšanju društva u cjelini.

U ovom će završnom radu biti riječi o rješavanju problema iz područja sigurnosti objekata i komunikaciji između mikrokontrolera, internet servera i korisnika u realnom vremenu. Samim time nameću se pitanja fizičkoga povezivanja podsustava, odabira odgovarajućih tehnologija te implementacija protokola za komunikaciju između odvojenih sustava. Rješavanjem ovih problema trebala bi se omogućiti pouzdana i brza komunikacija međusobno odvojenih sustava. Svrha toga je povezivanje podsustava u cjelinu kako bismo mogli komunicirati, razmjenjivati informacije ili slati naredbe jedni drugima. Bez ovakvog rješenja, pojedini uređaji morali bi raditi autonomno, a samim time korisnik nema pristup sustavu u realnom vremenu.



Slika 1: Komunikacija između uređaja.

Izvor: autor

## 2. Cilj rada istraživanja

Cilj je rada izrada sustava pomoću Arduino mikrokontrolera koji će omogućiti korisnicima da s udaljenoga mjesta, putem mobilnoga uređaja ili internet aplikacije, imaju nadzor nad prostorijama, kao što je detekcija pokreta, razni detektori koji mogu spriječiti ili upozoriti vlasnika od neželjenih događanja, poput poplave, požara ili nekog drugog fizičkog čimbenika. Na tržištu već postoje razni slični sustavi, no čest je problem nemogućnost komunikacije preko interneta, što podrazumijeva nemogućnost upravljanja sustavom i nemogućnost upozoravanja korisnika na neželjene događaje koji se događaju u udaljenome objektu. Rješavanjem navedenih problema uz pomoć Arduino uređaja, nekoliko različitih senzora i mobilnog uređaja, nastoji se omogućiti jeftina, a opet pouzdana i brza komunikacija između objekta i korisnika. Implementacijom Android ili iOS<sup>1</sup> aplikacije sustav bi trebao omogućiti komunikaciju između mobilne aplikacije i baze podataka u kojoj su spremljeni zadnji podatci o stanjima senzora. Na taj bi se način omogućilo da mobitel upozorava korisnika vibracijom i zvukom zvona na neželjeno stanje u objektu ili nekim drugim informacijama, kao što su temperatura u objektu, vlaga itd. Problem u komunikaciji može zadavati internetska veza u objektu, na lokaciji gdje je smješten internet server ili kod korisnika. Zbog dodatne sigurnosti trebalo bi istražiti i komunikaciju SMS<sup>2</sup> porukom između sustava u objektu i korisnika.

---

<sup>1</sup> iOS (prije iPhone OS) ime je za mobilni operacijski sustav koji je razvila američka tvrtka Apple Inc.  
<sup>2</sup> SMS (engl. *Short Message Service*) usluga je slanja kratkih tekstualnih poruka unutar GSM standarda mobilne telefonije.

### 3. Opis korištenih tehnologija

U ovom su poglavlju opisane tehnologije korištene pri izradi upravljačkoga i sigurnosnoga web sustava za nadzor prostorija. Cilj je poglavlja pružiti objašnjenja pojedinih tehnologija, kao i informacije vezane uz njihovo korištenje na projektu završnoga rada.

#### 3.1. Mikrokontroler

Mikrokontroler je digitalni elektronički uređaj u obliku integriranih sklopova čija je svrha kontroliranje uređaja i procesa. Mikrokontroler nosi integrirani mikroprocesor, memoriju, digitalne i analogne ulaze i izlaze, digitalne satove (engl. *timeri*), brojače, oscilatore, komunikacijske krugove i razne druge aditive. Mikrokontroler radi normalno u kontrolnoj petlji, dakle očitava ulaze i zatim podešava izlaze u skladu sa svojim programom. Petlja se stalno ponavlja dok traje kontrola procesa [1].

#### 3.2. Arduino

Arduino je mali sklop na ploči s USB priključkom za spajanje na računalo, serijskim i analognim pinovima koji su namijenjeni za spajanje na razne motore, releje, diode te razne digitalne i analogne senzore. Arduino može raditi s napajanjem preko USB<sup>3</sup> računala ili uz pomoć posebnog napajanja [2]. Za početak rada potrebno je napisati aplikaciju i učitati je u memoriju Arduina.

##### 3.2.1. Osnovne funkcije Arduina `setup()` i `loop()`

Osnovna se struktura programa za Arduino sastoji od dvije glavne funkcije, to su `setup()` i `loop()`. Funkcija `setup()` poziva se uvijek na početku pokretanja programa. U njoj se inicijaliziraju varijable, pokreću korišteni dodaci i slično. Funkcija `setup()`

---

<sup>3</sup> USB (engl. *Universal Serial Bus*) je rješenje za komunikaciju računala s vanjskim uređajima.



pokreće se samo jednom nakon uključivanja ili resetiranja Arduino uređaja. Funkcija *loop()* sadržava programski kod koji se uvijek ponovo ponavlja. Svako ponavljanje *loop()* funkcije može imati drugačiji ishod koji ovisi o više različitih čimbenika, npr. stanje ulaza na Arduino, vremenska ovisnost, provjera primljenih podataka putem mreže itd [2]. Slijedi jednostavan primjer koda na kojem se može vidjeti struktura programa i uključivanje pinova na Arduino. Prikazan je kod koji prikazuje osnovne funkcije Arduino programa i način na koji se može upravljati pinovima, u ovom slučaju pinom broj 13 koji se isključuje i uključuje u razmacima od jedne sekunde.

```
//Ova funkcija se pokreće samo jednom kod paljenja uređaja
void setup() {
    // initialize digital pin 13 as an output.
    pinMode(13, OUTPUT);
}

//Funkcija loop se ponavlja cijelo vrijeme
//i naš kod nalazi se u ovoj funkciji.
void loop() {

    digitalWrite(13, HIGH);    //aktiviranje izlaza
    delay(1000);               //čekaj sekundu
    digitalWrite(13, LOW);     //gašenje izlaza
    delay(1000);
}
```

U primjeru su pozvane dvije najčešće korištene funkcije *pinMode* i *digitalWrite*. Korištenjem funkcije *pinMode* s prvim parametrom zadaje se broj pina koji će se koristiti, dok drugi parametar postavlja pin kao ulaz ili izlaz (engl. *input/output*).

Funkcija *digitalWrite* ima također dva parametara, prvi parametar odnosi se na pin kojim se želi upravljati, a drugi se parametar odnosi na aktivaciju ili deaktivaciju zadanog pina. Funkcija *digitalWrite* uvijek se poziva u *loop* funkciji [2].

### 3.2.2. Varijable Arduino sustava

Kod mikrokontrolera ATmega328 važno je pravilno deklarirati varijable jer deklaracijom varijable mikrokontroler zna koliko memorije mora rezervirati za određenu varijablu. S dobrom deklaracijom varijabli može se uštedjeti na memoriji koje nema puno, ali zato istodobno moramo paziti i na problem preljeva. Problem preljeva javlja se kada, primjerice, u varijablu deklariranu kao „int“ pokušava upisati veći broj od 32767. Više o varijablama može se vidjeti na slici 2.

TIP	Veličina (bit)	Veličina (byte)	Minimalna vrijednost	Maksimalna vrijednost
unsigned byte	8	1	0	255
byte	8	1	-128	127
unsigned int	16	2	0	65535
int	16	2	-32768	32767
unsigned long	32	4	0	4294967295
long	32	4	-2147483648	2147483647

Slika 2: Prikaz raspoloživih varijabli i njihove vrijednosti.

Izvor: autor

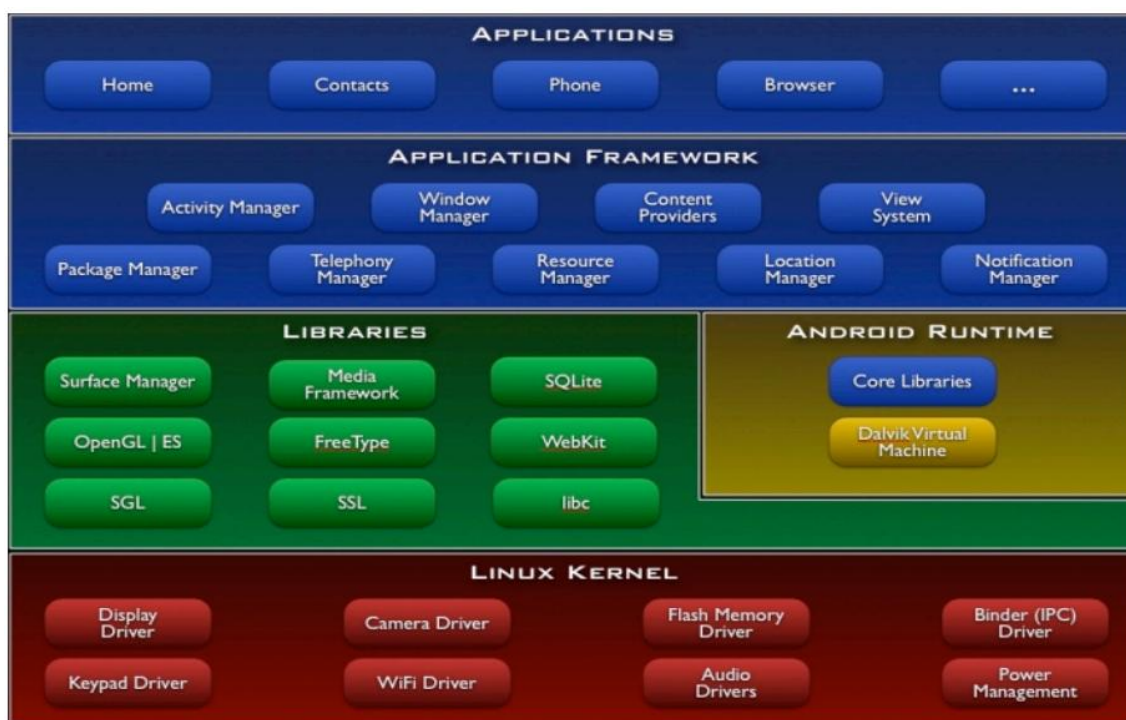
Da bismo koristili varijablu, potrebno ju je deklarirati. Postoje dva načina deklariranja:

```
int inputVariable1;
int inputVariable2 = 0;
```

Prvi način prikazuje deklaraciju, a drugi deklaraciju i pridruživanje. Varijablu je moguće deklarirati unutar funkcije *setup()* te će tada ona biti globalna varijabla. Ako nema potrebe da varijabla bude globalna, varijabla se može deklarirati unutar neke funkcije. Pridruživanje vrijednosti varijabli može se također zadati nekom funkcijom, kao u primjeru koji je u nastavku `int inputVariable1= analogRead(4);`.

### 3.3. Android

Google je 2005. godine kupio Android, Inc., malu kompaniju sa sjedištem u gradu Paolo Alto. Malo se znalo o poslovima koje je u to vrijeme obavljao Android, Inc., osim da su radili softver za mobilne telefone. Nakon kupnje Androida, Google planira ući u tržište mobilnih telefona. Googleov je tim razvio operacijski sustav za mobilne telefone temeljen na Linuxu 2.6 u C/C++ programskom jeziku koji koristi SQLite DBMS za pohranu podataka [3]. Arhitekturu Androida moguće je promatrati kao jedan stog koji sadrži više razina (prikazano na slici 3). Zbog otvorenoga sustava, programerima je izuzetno jednostavno pisati aplikacije koje koriste hardver uređaji uz pomoć besplatne inačice Android razvojnog paketa – ADK (engl. *Android development kit*) i Android virtualne mašine ADT (engl. *Android development tools*) u skladu s Eclipse IDE koji je korišten pri izradi ovoga rada. Android sustav je proizvođačima mobilnih telefona predstavljen kao fleksibilan i lako nadogradiv sistem [4].



Slika 3: Prikaz arhitekture Android sustava.

Izvor: <http://maksim.golivkin.eu/blog/2012/08/26/where-does-Android-fragmentation-hide/>

### 3.3.1. Android SDK

Android SDK<sup>4</sup> je paket softverskih razvojnih alata koji pruža podršku za razvoj, testiranje, pronalaženje i uklanjanje pogrešaka (engl. *debugging*) aplikacija. Najčešće korišten alat za izradu Android aplikacija je Eclipse IDE<sup>5</sup>, naziva ADT (engl. *Android Development Tools Plugin*) koji omogućuje jednostavan pristup (engl. *LogCat*), uređivaču datoteke *AndroidManifest.xml*, kontrolu dolaznih poziva, SMS simulaciju i slično [5].

Eclipse je besplatno višezjezično okruženje za razvoj programa koje se sastoji od integriranoga razvojnog okruženja (engl. *Software Development Environment*). Izgled razvojnoga alata Eclipse i emulatora prikazan je na slici 4. Može se koristiti za razvoj aplikacija u Javi i drugim programskim jezicima putem različitih softverskih dodataka, kao što su programski jezici: C, C++, COBOL, Perl, PHP, Python i drugi [6].

Android SDK emulator omogućava razvoj i testiranje aplikacija, a funkcionalnost operativnoga sustava skoro je jednaka Android operativnom sustavu koje se koriste na mobilnim uređajima. Jedino je značajno ograničenje emulatora nemogućnost uspostave dolaznih i odlaznih poziva. Korištenjem ADT-a, kao programskog dodatka za Eclipse, emulator se automatski poziva prilikom izvođenja, pronalaženja procesa i uklanjanja programskih pogrešaka (engl. *Debugger*).

DDMS (engl. *Dalvik Debug Monitoring Service*) – služi za kontrolu i nadzor pronalaženja i uklanjanja pogrešaka u aplikacijama. Cijeli se alat ponaša kao poveznica između IDE-a i aplikacija koje se izvršavaju na uređaju. Kao što je već spomenuto, na sustavu Android, svaka aplikacija pokreće se kao zasebni proces koji se izvodi kao zasebna instanca virtualnog stroja (Android emulatora). Prilikom pokretanja, DDMS se spaja na ADB<sup>6</sup> koji mu proslijeđuje informaciju o tome je li uređaj spojen ili nije. Kada je uređaj spojen pokreće se VM (engl. *Virtual Machine*), nadzorna usluga između ADB-a i DDMS-a, čija je uloga obavješćavanje DDMS-a je li virtualni stroj na uređaju pokrenut ili ugašen. Prilikom pokretanja virtualnog stroja aplikaciji se dodjeljuje

---

4 SDK je skraćenica engleskog izraza *software development kit* i to je tipični skup računarskih alatki koje programer koristi da bi napravio računarski program za određeni softverski paket.

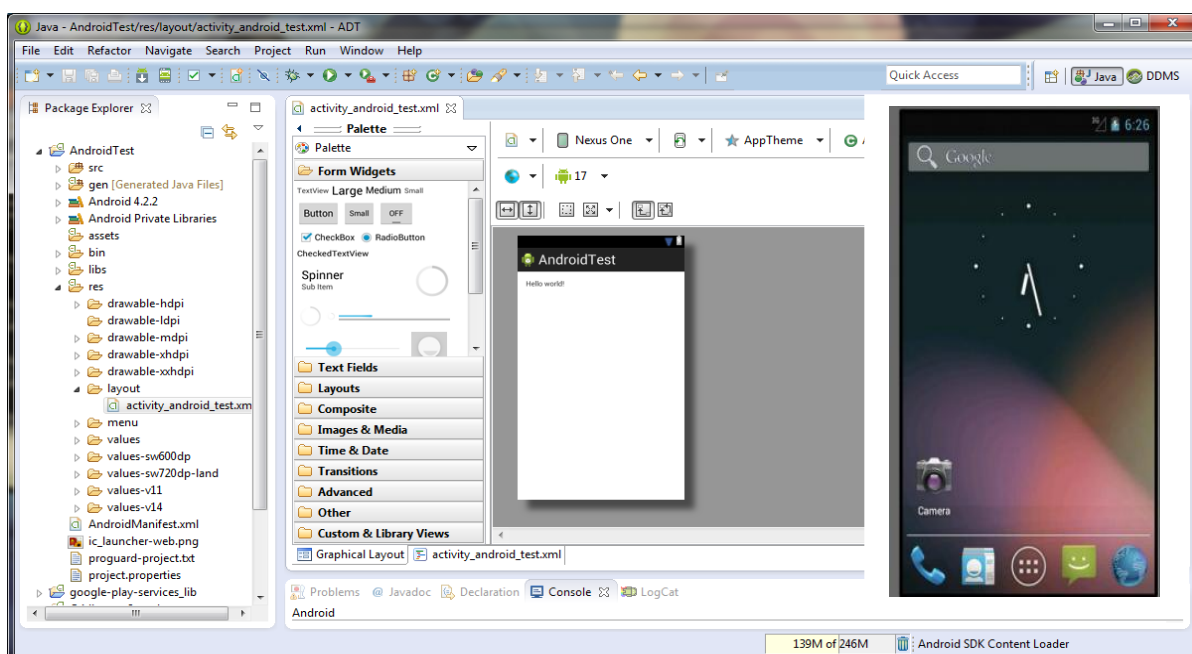
5 IDE (engl. *Integrated development environment*) razvojno okruženje za izradu programa.

6 ADB (engl. *Android Debug Bridge*) – klijentsko-poslužiteljska aplikacija.

jedinstveni ID i otvara vezu prema programu za pronalaženje i uklanjanje pogrešaka preko ADB-a (engl. *ADB Deamon*) na uređaju. Tako ostvarena veza između DDMS-a i virtualnog stroja radi preko WIRE<sup>7</sup> protokola [7].

AAPT (engl. *Android Asset Packaging Tool*) – koristi se za stvaranje i distribuciju Androidovoga programskog paketa u .apk formatu.

ADB (engl. *Android Debug Bridge*) – klijentsko-poslužiteljska aplikacija za instaliranje i pokretanje datoteka .apk na emulatoru ili uređaju te pristup komandnom retku uređaja. Koristi se i za povezivanje standardnoga programa za pronalaženje i uklanjanje pogrešaka (engl. *Debugger*) s kodom na emulatoru ili uređaju [8].



Slika 4: Prikaz Eclipse razvojnog alata, a na desnoj se strani nalazi Android emulator.

Izvor: autor

---

<sup>7</sup> WIRE se koristi za umrežavanje računala, raznih uređaja, odnosi na komuniciranje i dobivanje podataka od točke do točke.

### 3.4. PHP

PHP (engl. *Hypertext Preprocessor*) je objektno orijentiran programski jezik, namijenjen prvenstveno programiranju dinamičkih web stranica. PHP je skriptni jezik ugrađen u HTML<sup>8</sup> (ugrađeni HTML kod ili engl. *HTML embadded*) čiji se kod izvršava na strani poslužitelja (engl. *server-side*), što znači da se svi PHP programi izvode na poslužitelju, a klijentu se potom šalje samo rezultat izvođenja. Na strani klijenta vidi se samo HTML kod bez PHP-a. PHP (engl. *open source*) je besplatan programski jezik. Svatko tko želi može koristiti izvorne PHP kodove pisane u C programskom jeziku, mijenjati ih po svojoj volji te tako dodavati nove funkcije PHP-u [9]. Ovaj programski jezik također podržava sve popularnije baze podataka, kao što su MySQL, PostgreSQL, dBase, Oracle, ODBC<sup>9</sup> itd. Primjer pozivanja PHP skripte izgled ovako:

```
<html>  
<?php echo "Ovo je php ispis.";?>  
</html>
```

#### 3.4.1. Prednosti PHP jezika

Prednosti PHP jezika su sljedeće:

1. performanse: PHP je vrlo brz – jedan relativno jeftin server može obraditi milijun zahtjeva dnevno;
2. proširivost: horizontalno sustav se može efikasno i jeftino proširiti na veliki broj običnih, jeftinih servera za podatke;
3. integriranost sa sustavom za upravljanje gotovo svim bazama podataka;
4. niska cijena: PHP je besplatan;
5. lako se uči: sintaksa PHP-a slična je sintaksama drugih programskih jezika, prvenstveno jezika C i Perl<sup>10</sup>;
6. prenosivost: PHP je na raspolaganju za mnoge operacijske sustave [10].

---

<sup>8</sup> HTML (*HyperText Markup Language*) je prezentacijski jezik za izradu web stranica.

<sup>9</sup> ODBC (*Open Database Connectivity*) je program za pristup sustavu za upravljanje bazama podataka.

<sup>10</sup> Perl je programski jezik opće namjene, a odlikuje ga kvalitetan repozitorij gotovih programskih rješenja.

### 3.5. MySql baza podataka

MySQL je sistem za upravljanje bazom podataka. Baza podataka je strukturno povezan skup podataka koji može biti od obične liste zaposlenika pa sve do goleme količine podataka u korporacijskim mrežama. Relacijska baza podataka je baza podataka koja podatke pohranjuje u odvojene tabele umjesto u jednu tabelu. Ovo omogućuje brži i fleksibilniji rad s bazama podataka. Tabele su povezane definiranim relacijama što omogućuje kombiniranje podataka iz nekoliko tabela. MySQL licenca je GLP (*General Public License*), besplatna za korištenje, ali isto tako postoji i MySQL Enterprise verzija programa koja nije besplatna i koja nudi naprednije opcije od osnovne licence. MySQL je prvobitno nastao kao interni projekt, a izvorni kod ove aplikacije napisan je u C i C++ programskom jeziku. Iz tih razloga ova aplikacija radi na različitim operativnim sustavima. Njezina popularnost usko je povezana s popularnošću PHP-a jer se često koriste zajedno. Nekoliko najprometnijih mrežnih stranica, kao što su Flickr, Facebook, Wikipediju i YouTube, koriste MySQL bazu za pohranu podataka i praćenje korisnika [11]. Na serveru može postojati veći broj potpuno samostalnih baza podataka. Jedna se baza obično sastoji od više tablica, dok se unutar jednog projekta može upravljati podacima iz više baza na serveru. Korisničko se sučelje ove baze podataka može koristiti preko internet preglednika na adresi <http://localhost/phpmyadmin/> (prikazan na slici 5) ili može koristiti druge programe kao što su MySQL Workbench, SQLyog, MySQL Editor, Visual GUI Tool i razne druge.

#### 3.5.1. Prednosti MySQL-a

Prednosti MySQL-a su sljedeći:

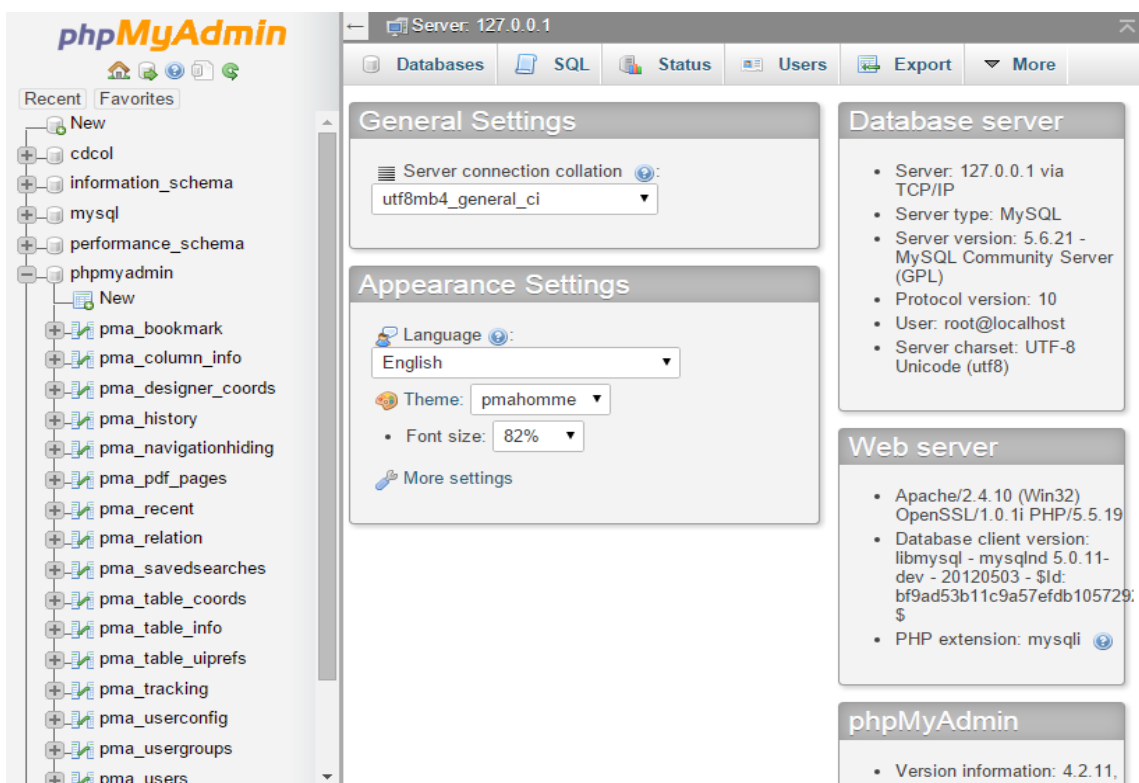
1. brzina (jedna od najbržih baza podataka, ako ne i najbrža);
2. pouzdanost;
3. mala zahtjevnost za sistemskim resursima;
4. fleksibilno poboljšavanje performansi;
5. rad na različitim platformama;
6. podržavanje velike količine programskih jezika;

7. podrška za ODBC;
8. besplatno ili povoljno licenciranje;
9. jaka podrška korisnika;
10. dostupnost izvornog koda [12].

### 3.5.2. Nedostaci MySQL baza podataka

Nedostaci MySQL baze podataka su sljedeći:

1. nedostatak nekih mogućnosti SQL-a koje neke druge baze podataka imaju;
2. komplicirano rukovanje izvornim kodom [12].



Slika 5: Prikaz phpMyAdmin sučelja za upravljanje bazom podataka.

Izvor: autor



## 4. Potrebni uređaji za realizaciju projekta

Za realizaciju projekta potreban je jedan ili više Arduino uređaja, ovisno o tome koliko je potrebno spojiti različitih senzora. Arduino uređaj nema mogućnost direktnog spajanja na internet ili na lokalnu mrežu te je tako potreban dodatak koji se ugradi na Arduino uređaj (prikazan na slici 7). Postoje dvije mogućnosti spajanja na internet, prva i jeftinija mogućnost je spajanje s internet uređajem preko kabla (engl. *Arduino Ethernet Shield*) ili preko uređaja koji podržava bežično povezivanje (engl. *Arduino Wireless Shield*).

Od senzora je potrebno sljedeće: analogni senzor za temperaturu *DHT11*, senzor za vodu analogni *HV3N Funduino*, Relej 5V s četiri sklopke, senzor vatre analogni, bežični (engl. *wireless*) radio prijemnik 433MHZ, bežični senzor pokreta.

### 4.1. Arduino

Arduino je elektronička platforma otvorenog koda. Bazira se na hardveru i softveru koji su fleksibilni i jednostavni za korištenje. Hardver Arduina je (engl. *open source*) što znači da na tržištu postoji mnogo klonova koji se mogu kupiti po znatno nižoj cijeni. Klon Arduina najlakše je prepoznati ako na ploči postoji natpis *Duino*, na primjer *Free Duino*, *DFR Duino*, *Black Duino*, *Mega Duino* itd [13]. Arduino Uno je mikrokontrolna ploča zasnovana na Atmega328 mikrokontroleru. Ima 14 digitalnih izlaznih/ulaznih pinova od kojih se 6 može koristiti kao PWM<sup>11</sup> izlaza, 6 analognih ulaza, 16 MHz keramički otpornik, USB priključak, priključak za napajanje i tipka za ponovno pokretanje uređaja. Arduino također sadržava sve što je potrebno kao podrška mikrokontroleru, kao na primjer jednostavno spajanje na računalu s USB kablom te mogućnost napajanja s DC adapterom ili baterijama. Za razliku od prethodnih Arduino ploča, Arduino Uno ne koristi FTDI USB<sup>12</sup> čip za serijsku komunikaciju, već umjesto toga Atmega 16u2 je programiran kao USB serijski pretvarač. Atmega 328 ima kapacitet memorije 32 KB (s 0,5 KB iskoristivih pri podizanju). Također ima i 2 KB

---

11 PWM (engl. *Pulse width modulation*) dekodira poruku pretvara u signal čitljiv računalu.

12 FTDI USB (engl. *Future Technology Devices International*) USB čip za serijsku komunikaciju.

SRAM<sup>13</sup> memorije i 1 KB EEPROM<sup>14</sup> memorije (može biti čitana i zapisivana s EEPROM knjižicom definiranom u programskom sustavu) [2]. Više o tehničkim karakteristikama nalazi se u tablici 1.

Editor za pisanje programa za Arduino također je besplatan i radi s većinom platformi, kao na primjer Windows, Linux, Mac. Arduino IDE<sup>15</sup> baziranim je na Javi i korisniku pruža jednostavno programiranje korištenjem unaprijed gotovih funkcija.



Slika 6: Prikaz Arduino uređaja.

Izvor: <http://arstechnica.com/information-technology/2013/10/arduino-creator-explains-why-open-source-matters-in-hardware-too/>

Tablica 1. Osnovne karakteristike Arduino uređaja.

Mikrokontroler	ATmega328
Radni napon	5V
Ulazni napon (preporučeno)	7-12V
Ulazni napon (ograničenje)	6-20V
Digitalni I/O Pinovi	14 (od čega 6 omogućava PWM izlaz)
Analogni Input Pinovi	6
DC Struja za I/O Pin	40 mA
DC Struja za 3.3V Pin	50 mA
Flash Memorija	32 KB (ATmega328) i 0.5 KB podizanju sustava
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Brzina procesora	16 MHz

Izvor: <http://arduino.cc/>

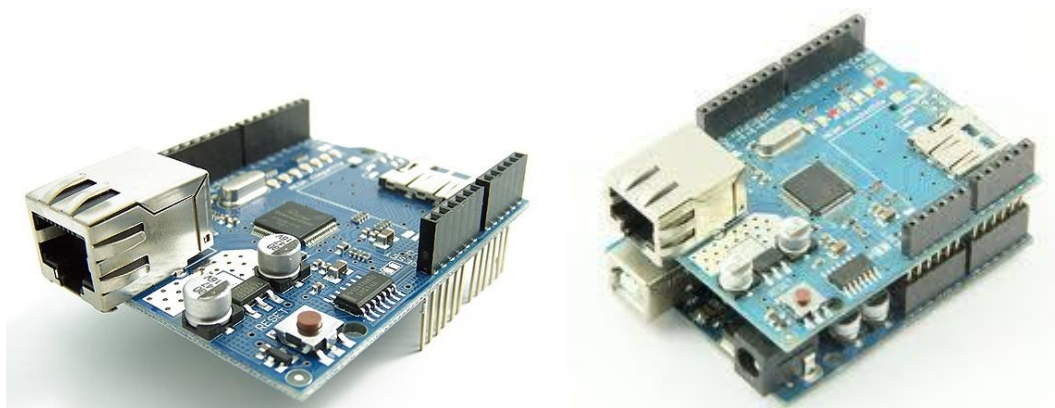
13 SRAM (engl. *Static Random Access Memory*) jedna je od izvedbi poluvodičke memorije.

14 EEPROM (engl. *Electrically Erasable Programmable Read-Only Memory*) je električno izbrisiva programibilna ispisna memorija.

15 IDE (engl. *Integrated development environment*) je razvojno okruženje za izradu programa.

## 4.2. Arduino Ethernet Shield

Arduino Ethernet Shield omogućava Arduino pločama da se povežu na internet. Zasnovan je na Wiznet W5100 ethernet čipu koji omogućava mrežnu komunikaciju koristeći TCP 16 i UDP 17 protokole. Podržava do četiri istovremena spajanja. Povezivanje s Arduino pločom izvedeno je tako da su svi pinovi na Arduino ploči jednaki priključnicima na Arduino Ethernet Shield ploči (prikazano na slici 2). Time je omogućeno dodavanje modula jednih preko drugih uz zadržavanje osnovnog rasporeda pinova. Ethernet Shield ima standardni RJ-45 mrežni priključnik s integriranim transformatorom i PoE<sup>18</sup> mogućnostima. Na ploči postoji i priključnik za micro-SD kartice koji se može koristiti za čuvanje datoteka koje će se slati preko mreže. Kompatibilna je s Arduino Uno i Mega pločama (uz korištenje Ethernet biblioteke) [14].



Slika 7: Prikaz Arduino Ethernet Shield uređaja.

Izvor: <http://waihung.net/arduino-ethernet-shield-r3/>

Za rad s micro-SD karticom koristi se SD biblioteka. Prva verzija ove ploče imala je SD priključnik pune veličine koji je zamijenjen micro-SD priključnikom. Ploča također posjeduje i kontroler za resetiranje koji omogućava ispravno resetiranje W5100 čipa, dok prethodna verzija Arduino Ethernet Shield nije bila kompatibilna s Mega pločama i

---

16 TCP (engl. *Transmission Control Protocol*) jedan od osnovnih protokola za razmjenu podataka putem interneta, protokol provjerava je li paket isporučen.

17 UDP (engl. *User Datagram Protocol*) jedan od osnovnih protokola za razmjenu podataka putem interneta, protokol ne provjerava je li paket isporučen.

18 PoE (engl. *Power over Ethernet*) – napajanje preko mrežnoga kabla.

morala se ručno resetirati nakon uključivanja. Ne isporučuje se s PoE modulom, već je zasebna komponenta koja se može dodatno dodati. Arduino komunicira s W5100 i SD karticom preko SPI<sup>19</sup> porta. Taj se port nalazi na pinovima 11, 12 i 13 na Arduinovim pločama, odnosno na pinovima 50, 51 i 52 na Mega ploči. Arduino Ethernet Shield i na koji se način spaja na Arduino može se vidjeti na slici 7.

Za pisanje programa koji koriste Arduino Ethernet Shield dostupna je i Ethernet biblioteka koja olakšava programeru korištenje raznih funkcija, kao što su (engl. *web request*, *telnet*, *UDP server/client*, *TCP server/client*).

Tablica 2. Osnovne karakteristike Arduino Ethernet Shield uređaja.

Arduino Ethernet Shield	karakteristike
Kompatibilnost	IEEE802.3af
Izlazni napon na pinovima	3.3V, 5V, 9V
Napon	36V-50V
Digitalni I/O Pinovi	14 (od čega 6 omogućava PWM izlaz)
Analogni Input Pinovi	6
DC Struja za I/O Pin	40 mA
DC Struja za 3.3V Pin	50 mA
Priključak	RJ-45

Izvor: <http://www.arduino.cc/en/Main/ArduinoEthernetShield>

U nastavku je primjer koda koji prikazuje kako komunicirati uz pomoć Arduino Ethernet Shield uređaja s internet stranicama ili internet aplikacijama (engl. *web request*). U navedenom primjeru funkcijom *setup* poziva se domena <http://www.google.com>, a funkcijom *loop* dobiva se odgovor od zatražene domene i ispisuje se u kontrolnoj ploči. Odgovor od zatražene domene je u obliku teksta, html-a, xml-a itd.

---

<sup>19</sup> SPI (engl. *Serial Peripheral Interface*) je komunikacijski sustav između mikroprocesora i drugih čipova.

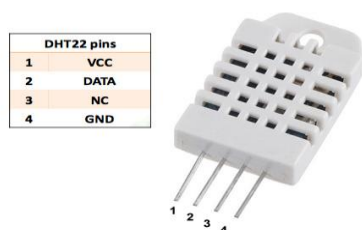
```
#include <SPI.h>
#include <Ethernet.h>
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
char server[] = "www.google.com";
IPAddress ip(192, 168, 0, 177);
EthernetClient client;

void setup() {
    // Otvaram serijsku komunikaciju i čekam da se port otvori:
    Serial.begin(9600);
    while (!Serial)
    {
        // cekam da se spoji na internet
    }
    if (Ethernet.begin(mac) == 0)
    {
        Serial.println("Greška kod postavke DHCP-a");
        Ethernet.begin(mac, ip);
    }
    // Ethernet shield treba 1 sekundu za inicijalizaciju
    delay(1000);
    Serial.println("spajam se...");
    if (client.connect(server, 80))
    {
        Serial.println("Spojeno");
    }
    else
    {
        Serial.println("Greška kod spajanja ");
    }
}

void loop()
{
    if (client.available())
    {
        char c = client.read();
        Serial.print(c);
    }
    if (!client.connected())
    {
        Serial.println("Greška kod spajanja.");
        client.stop();
    }
}
```

### 4.3. Senzor za temperaturu DHT11

Navedeni senzor mjeri temperaturu i vlažnost zraka i ima oznaku DHT11 ili nešto kvalitetniji senzor nosi oznaku DHT22. DHT11 mjeri temperaturu u opsegu od 0 do 50 stupnjeva Celzijusa s preciznošću od  $\pm 2$  stupnja te vlažnost u opsegu od 20 do 80% s greškom 5%. Senzor se može vidjeti na slici 8. Senzor radi na 5V, a izlaz koji daje je digitalni ili analogni. DHT22 je kvalitetniji senzor koji mjeri temperaturu u opsegu od -40 do +125 stupnjeva Celzijusa, uz preciznost 0.5 stupnjeva [15].



Slika 8: Prikaz DTH11 senzora za temperaturu i pinove za spajanje na Arduino.

Izvor: <http://electronica.mercadolibre.com.mx/otros/sensor-de-temperatura-y-humedad-dht11-para-arduino>

### 4.4. Senzor za vodu HV3N (Funduino)

Senzor za vodu HV3N dizajniran je za detekciju vode koji se može koristiti kao senzor padalina ili za detekciju vode u prostorijama. Uređaj se sastoji od tri dijela: ploče s priključkom, A1 MQ otpornika i nekoliko redaka golih žica. Ovaj senzor radi na 5V i vraća signal analognim putem. Senzor, ako je u cijelosti uronjen u vodu, vraća 5V. Iz tog je razloga bitno da ne postoji unutarnji otpor u cijelom sistemu. Prva naznaka da postoji vlaga je 0,1V na izlazu iz senzora. Za navedeni senzor Arduino učitava stanje od 0 do 1000. Maksimalno očitavanje je kada je senzor uronjen u vodi [16].



Slika 9: Prikaz DTH11 senzora za vodu.

Izvor: [http://www.aliexpress.com/arduino-water-sensor\\_reviews.html](http://www.aliexpress.com/arduino-water-sensor_reviews.html)

#### 4.5. Releji ulaz 5V

Releji radi na visokom naponu izmjenične struje do maksimalno 250V. Najčešće se koriste u interaktivnim projektima, a može se koristiti za kontrolu rasvjete, električne i druge eksperimente. Upravljanje se vrši s istosmjernom strujom 5V [17].



Slika 10: Prikaz pločice s 8 releja.

Izvor: <http://arduino-info.wikispaces.com/ArduinoPower>

#### 4.6. Bežičan wireless senzor pokreta

Senzor radi putem bežičnoga signala na frekvenciji od 443MHZ ili na frekvenciji od 315MHZ, ovisno o proizvođaču. Visina montaže trebala bi biti od 1.8 do 2.5 metara iznad tla. Maksimalna udaljenost koju senzor može očitati, prema tvorničkim uputama je do 10 metara, dok radna udaljenost i komunikacija s bežičnim prijemnikom doseže čak do 300 metara na otvorenom. Raspon vidljivosti je 120 stupnjeva, a za napajanje koristi 9V bateriju koja bi, ovisno o proizvođaču, trebala trajati u razdoblju od 6 do 12 mjeseci [18].

#### 4.7. Senzor vatre

Senzor vatre služi za detekciju požara ili otvorene vatre, a također može poslužiti i za detekciju ili jačinu svjetla. Kut detekcije vatre počinje od 60 stupnjeva, dok plamen upaljača može otkriti već od 760 do 1100 cm. Kada je intenzitet vatre veći, tada se i udaljenost detekcije uvećava. Radni napon iznosi od 3V do 5V [19].

## 5. Praktični rad

Praktični se rad sastoji od triju različitih sustava koji komuniciraju preko internetske veze. Arduino uređaj uz pomoć mrežne utičnice (engl. *Arduino Ethernet Shield*) komunicira s udaljenim internet serverom te tako sprema podatke o stanjima senzora u MySQL bazu podataka, nakon toga Android aplikacija uzima podatke iz MySQL baze podataka i informira korisnika o mogućoj prijetnji u objektu.

### 5.1. Arduino aplikacija

Arduino aplikacija odnosi se na aplikaciju koja se učitava u sam Arduino uređaj, može biti maksimalne veličine 16 KB. Aplikacija se sastoji od 12 glavnih funkcija koje služe za skupljanje podataka od senzora za slanje i primanje podataka s udaljenoga internetskog poslužitelja. Svih 12 spomenutih funkcija koriste i ostale podfunkcije i razne gotove knjižice za koje postoji dokumentacija na adresi: <http://www.arduino.cc/en/Reference>.

#### 5.1.1. Funkcija setup

Funkcija *setup* glavna je funkcija bez koje Arduino uređaj ne može raditi. Služi za inicijaliziranje varijabli, pokretanje funkcija koje se koriste samo jednom prilikom pokretanja uređaja, a isto tako važno je ispravno konfigurirati pinove za primanje ili slanje signala tako da se upiše *pinMode(2,INPUT);* ili *pinMode(2,OUTPUT);*. Primjer *setup* funkcije prikazan je na slici 11. U funkciji *pinMode()* prvi parametar služi za inicijaliziranje željenog pina, a drugi parametar zadaje pinu način rada *input/ output* [2].

```
//GLAVNA FUNKCIJA KOD ARDUINO
void setup() {
  Serial.begin(9600);
  pinMode(2, INPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(8, OUTPUT);
  //POKREĆEM PRIMANJE PODATAKA ZA BEZŽIČNI SENZOR POKRETA
  mySwitch.enableReceive(0);
  delay(7000);
  Serial.begin(9600);
  while (!Serial) {}
  if (Ethernet.begin(mac) == 0) {
    Serial.println("NE MOGU KONFIGURIRATI DHCP");
    Ethernet.begin(mac);
  }
  Serial.print("MOJA IP ADRESA JE: ");
  Serial.print(Ethernet.localIP());
}
```

Slika 11: Prikaz funkcije *setup*.

Izvor: autor



### 5.1.2. Funkcija *loop*

Funkcija *loop* neophodna je funkcija za ispravan rad uređaja i pokreće se odmah nakon funkcije *setup*. Funkcija *loop* radi upravo ono što joj samo ime sugerira, uzastopno se vrti i izvršava kod koji je upisan u tijelu funkcije. Zadaća programera je da svoju aplikaciju prilagodi takvome načinu rada, mora dobro razmisliti i logično razdijeliti koje senzore ili funkcije treba provjeravati svaku milisekundu, a koje svaku sekundu ili više. Kao primjer takve logike mogu se spomenuti dvije funkcije u *loop* funkciji, senzor pokreta koja se provjerava svakih 100 milisekundi ili funkcija koja šalje podatke udaljenom serveru svakih 10 sekundi.

```
int brojacl=0;
void loop() {
    //OVA FUNKCIJA IMA NAJVEĆU FREKVENCIJU PROVJERE
    SenzorPokreta();
    //OSTALE FUNKCIJE PROVJERAVAM NAKON ŠTO SENZOR POKRETA PROVJERIM 20000
    if(brojacl==20000){
        delay(2000);
        UzmiTemperaturu();
        ProvjeraPoplave();
        UzmiVrijednostiDimaVatre();
        PokreniWebRequest();
        PaliGasiBojler();
        PaliGasiAlarm();
        PaliGasiRelay();
        Serial.println("broj:");
        Serial.println(brojacl);
        brojacl=0;
    }
    brojacl++;
}
```

Slika 12: prikazuje *loop* funkciju.

Izvor: autor

### 5.1.3. Funkcija *WebRequest*

Funkcija *WebRequest* služi za komunikaciju između Arduino uređaja i udaljenog servera, u ovom primjeru za komunikaciju sa serverom na domeni <http://www.arduino.pc1.hr/kontroler.php>. Funkcija podatke i stanja senzora šalje putem POST metode. U ovoj je funkciji potrebno ispravno popuniti PORT preko kojega se

šalju podaci (u ovom slučaju 80), ime servera i naslov poruke (zvani *message header*) koje se šalje prema željenom serveru. Također, bitno je napomenuti da ova funkcija koristi gotove knjižice koje pomažu pravilno razdijeliti svaku poruku, upakirati poruku u TCP omotnicu i poslati prema ruteru. Iza svakog zahtjeva prema serveru (engl. *WebRequest*) dobiva se i odgovor s udaljenog servera (engl. *Web Response*) pa tako i u ovome slučaju. Funkcija *WebResponse()* nije prikazana u ovome radu jer sadrži previše linija koda. Ukratko, funkcija *WebResponse()* provjerava je li od servera dobivena ispravna poruka. Ako je odgovor s udaljenog servera ispravan, poruka izgleda ovako: *Success 200 OK* ili ako se dogodila greška na serveru dobivamo poruku *Not found 404*, *Not implemented 501*, *Forbidden 403*, *Internal Error 500*. Ako se nije dogodila greška na udaljenom serveru, dobivamo sadržaj koji generira PHP aplikacija *kontroler.php*, a sadržaj sadržava željene korisničke postavke koje se nalaze u bazi podataka, npr. taj odgovor može izgledati ovako: *\$20\$0\$0\$0\$1\$1\$5*. Svaki znak *\$* razdjeljuje jednu vrijednost pa tako prva vrijednost prikazuje željenu temperaturu, druga treća i četvrta aktivnost releja, peta aktivnost alarma, šesta aktivnost sirene (u slučaju ako postoji nedozvoljeno stanje) i sedmo brzina osvježavanja i sinkronizacije u sekundama Arduino uređaja i udaljenog servera.

```
void WebRequest() {
    //ako je uspjela konekcija
    if (client.connect(server, 80)) {
        String data="arduino_request="+String(temperatura)+"&"+
        String(vlaga)+"&"+String(motion_status)+"&"+
        String(VratiAlarmZaVodu)+"&"+String(dimVatraPrisutni)+"&";
        client.println("POST /kontroler.php HTTP/1.1");
        client.println("Host: arduino.pcl.hr");
        client.println("User-Agent: arduino-ethernet");
        client.println("Content-Type: application/x-www-form-urlencoded");
        client.print("Content-Length: ");
        client.println(data.length());
        client.println("");
        client.print(data);
        lastConnectionTime = millis();
    } else {
        brojac1=0;
        Serial.println("connection failed");
        Serial.println("disconnecting.");
        client.stop();
    }
}
```

Slika 13: Prikaz *WebRequest* funkcije.

Izvor: autor

### 5.1.4. Funkcija PaliGasiAlarm

Glavna je zadaća *PaliGasiAlarm* funkcije da provjerava stanje varijabla u koje se zapisuje stanje senzora svakih nekoliko milisekunda. Ako postoji stanje u određenoj varijabli koje ne odgovara korisničkim postavkama ili stanje koje može biti štetno (vatra, poplava, dim, plin ili pokret za vrijeme kada je alarm upaljen), poziva se funkcija *digitalWrite(8,1)* te tako aktivira pin 8 koji aktivira sirenu i dodjeljuje varijabli *aktivnost\_alarma* vrijednost 1 koja u funkciji *WebRequest()* poziva korisnika na mobitel i ukazuje na mogućnost nedozvoljenog stanja u objektu. Funkcija se može vidjeti na slici 14.

```
void PaliGasiAlarm()
{
    if((set_sirena=="1" && set_alarm_aktivnost=="1"
    && motion_status=="1" && aktivni_alarm==0) ||
    (set_sirena=="1" && dimVatraPrisutni==1 && aktivni_alarm==0) ||
    (set_sirena=="1" && VratiAlarmZaVodu=="1" && aktivni_alarm==0))
    {digitalWrite(8, 1); aktivni_alarm=1;}
    else if((dimVatraPrisutni==0 && VratiAlarmZaVodu=="0" && aktivni_alarm==1
    && (set_alarm_aktivnost=="0" && motion_status=="0")) || set_sirena=="0" )
    {digitalWrite(8, 0); aktivni_alarm=0;}
    if(set_alarm_aktivnost=="0"){ motion_status="0";}
}
```

Slika 14: Prikaz *PaliGasiAlarm* funkcije.

Izvor: autor

### 5.1.5. Provjera poplave i analogno učitavanje podataka

U funkciji *ProvjeraPoplave* koristi se analogno učitavanje vrijednosti. Na slici 15 prikazana je sintaksa za dobivanje analogne vrijednosti iz senzora pomoću funkcije *analogRead*.

```
void ProvjeraPoplave() {
    int moisture = analogRead(1);
    if(moisture > 300) {
        VratiAlarmZaVodu="1";
    } else if(moisture<=300) {
        VratiAlarmZaVodu="0";
    }
}
```

Slika 15: Prikaz funkcije *ProvjeraPoplave*.

Izvor: autor

### 5.1.6. Provjera pokreta

Aplikacija podržava dvije mogućnosti provjere pokreta. Prva je mogućnost žični senzor pokreta, a druga bežični senzor pokreta. Kod provjere stanja bežičnoga senzora, korištena je gotova knjižica *RCSwitch.h* koja se može preuzeti sa sljedećega linka: <https://code.google.com/p/rc-switch>. Više o kodu može se vidjeti na slici 16.

```
void SenzorPokreta(){
  if (mySwitch.available()) {
    int value = mySwitch.getReceivedValue();
    if (value == 0) {
      Serial.print("Unknown encoding");
    } else {
      Serial.print("Received ");
      Serial.print( mySwitch.getReceivedValue() );
      Serial.print(" / ");
      Serial.print( mySwitch.getReceivedBitlength() );
      Serial.print("bit ");
      Serial.print("Protocol: ");
      Serial.println( mySwitch.getReceivedProtocol() );
      motion_status="1";
    }
    mySwitch.resetAvailable();
  }
}

void SenzorPokretaZicni(){
  vrijednost_pir_senzora = digitalRead(2);
  if (vrijednost_pir_senzora == HIGH) {
    delay(150);
    if (pirState == 0) {
      Serial.println("Motion detected!");
      motion_status="1";
      pirState = HIGH;
    }
  } else {
    delay(150);
    if (pirState == HIGH){
      Serial.println("Motion ended!"); pirState = LOW;
    }
  }
  delay(250);
}
```

Slika 16: Prikaz dviju funkcija za bežični i žični senzor pokreta.

Izvor: autor

## 5.2. Android aplikacija

Android aplikacija namijenjena je uređajima koji rade na platformi Android. Nakon instalacije aplikacije u telefon korisnika, automatski se instalira proces koji je stalno aktivan bez obzira je li primarna aplikacija (korisničko sučelje) aktivno ili ne. Proces ima zadaću da za vrijeme aktivnosti alarma provjerava stanje senzora i obavijesti korisnika o mogućem požaru, poplavi ili neovlaštenom kretanju po objektu. Pozadinski proces Android aplikacije obavještava korisnika na način da aktivira vibraciju i zvono telefona. Isto tako, kreira obavijest na vrhu ekrana s opisom o kojem se problemu radi. Za rad aplikacije potrebna je internetska veza jer svako stanje senzora ili promjena uzrokovana korisničkim zahtjevom zapisuje se u MySQL bazu koja je smještena na udaljenom serveru. Korisničko sučelje Android aplikacije može se vidjeti na slici 17.

Trenutna temperatura je: 19.00 C  
Trenutna vlaga u prostoriji je: 55.00 %  
Zadnja promjena: 05.02.2015 13:32:52

**Postavke sustava**

–  + 20 C je željena temperatura u prostoriji

☒ ON Garažna vrata otvori zatvori

☒ ON Prekidač 1

☒ ON Prekidač 2

☐ OFF Aktivnost alarma

☒ ON Aktivnost sirene

**Dodatne postavke**

C° Od sati:  H Dnevna temperatura

C° Od sati:  H Noćna temperatura

Slika 17: Prikaz početne forme Android aplikacije.

Izvor: autor

U nastavku je prikazan kod Android aplikacije koji se spaja na udaljeni web server, uzima zadnje stanje senzora iz MySQL baze podataka i prosljeđuje dobivene podatke metodi *ObavijestiKorisnika*. Zadaća metode *ObavijestiKorisnika* je da putem parametra dobije zadnje stanje senzora, provjeri postoji li potencijalna opasnost od nedozvoljenog stanja u objektu, i ako postoji, da aktivira vibraciju, zvono telefona i ispiše o kakvoj se prijetnji radi.

```
private Runnable timerRunnable = new Runnable() {
    public void run() {
        new Thread(new Runnable() {
            @Override
            public void run() {
                while(true)
                {
                    String odgovor=
                    Funk.WebUpitOdgovor
                    ("http://arduino.pc1.hr/device.php");
                    String arrOdgovor[]=odgovor.split("x");
                    if(arrOdgovor.length==4)
                    {
                        ObavjestiKorisnika(arrOdgovor);
                    }
                    else
                    {
                        Log.d("Greška web odgovora: ", odgovor);
                    }
                }
            }
        }).start();
    }
};
```

```
private Boolean UpozoroSam=false;
private void ObavjestiKorisnika(String arrOdgovor[])
{
    String dim,voda,pokret,aktivnost_alarma;
    dim=arrOdgovor[0];
    voda=arrOdgovor[1];
    pokret=arrOdgovor[2];
    aktivnost_alarma=arrOdgovor[3];

    if((dim.equals("1") || voda.equals("1") ||
    (aktivnost_alarma.equals("1") && pokret.equals("1")))) &&
    !UpozoroSam)
```

```
{
    UpozoroSam=true;
    String poruka="",naslov="";
    if(dim.equals("1"))
    {naslov="Detektiran dim";
    poruka="Detektirao se senzor dima, svakako provjerite"+
    " dali je u objektu sve u redu.";}
    if(voda.equals("1"))
    {naslov="Detektirana voda";
    poruka="Senzor vode učitava vodu, poplavu.";}
    if(pokret.equals("1"))
    {naslov="Detektiran pokret";
    poruka="Vaš alarm je upaljen ali netko se kreće po
    objektu";}
    Intent intent = new Intent(this, Nalozi.class);
    PendingIntent pIntent = PendingIntent.getActivity(this,
    0,intent, 0);
    NotificationCompat.Builder mBuilder =
        new NotificationCompat.Builder(this);
    mBuilder.setSmallIcon(R.drawable.icon);
    mBuilder.setContentTitle(naslov);
    mBuilder.setContentText(poruka);
    Uri uri= Uri.parse("Android.resource://" +
    getPackageName() + "/" + R.raw.elegant_ringtone);
    mBuilder.setSound(uri);
    NotificationManager mNotificationManager =
        (NotificationManager) getSystemService(
        Context.NOTIFICATION_SERVICE);

    mNotificationManager.notify(1,mBuilder.setContentIntent(pIntent
    )
        .setAutoCancel(true)
        .build()
    );
    Vibrator v =
    (Vibrator) this.getSystemService(Context.VIBRATOR_SERVICE);
    v.vibrate(20000);

    }
    else if(dim.equals("0") || voda.equals("0") ||
    pokret.equals("0") && UpozoroSam)
    {
        UpozoroSam=false;
    }
}
```

### 5.3. Internet aplikacija

Internet aplikacija ima dva bitna zadatka. Prvi je zadatak da prilikom dohvaćanja stanja senzora putem POST naredbe provjeri stanje svakog senzora. Nakon toga sprema u bazu podataka zadnja stanja senzora, osim ako nisu potpuno jednaka kao i prethodna stanja koja su već zapisana u bazi. Aplikacija također vraća zadnja stanja senzora koja su bitna Android aplikaciji i Android pozadinskom procesu, koji svakih 5 sekundi provjerava postoji li naznaka za alarmiranje korisnika. Drugi zadatak internet aplikacije je da korisnik iz bilo kojeg internet preglednika može pristupiti korisničkom sučelju na kojem može vidjeti temperaturu i vlagu u objektu, upaliti ili ugaziti bojler za grijanje, upravljati raznim prekidačima, aktivirati ili deaktivirati alarm u udaljenom objektu. Korišteni alati za ovu aplikaciju su PHP, Java Script, Ajax i MySQL baza podataka. Prikaz aplikacije može se vidjeti na slici 18. Ova internet aplikacija trenutno je smještena na domenu <http://arduino.pc1.hr/>.

The screenshot displays two panels of the web application interface. The left panel, titled 'Postavke sustava' (System Settings), shows current sensor data: 'Trenutna temperatura je: 19.00 C', 'Trenutna vlaga u prostoriji je: 55.00 %', and 'Zadnja promjena: 05.02.2015 13:32:52'. Below this, it lists system settings: a target temperature of 20 C with minus/plus controls, a garage door status (ON), two switch statuses (ON), alarm status (OFF), and siren status (ON). The right panel, titled 'Dodatne postavke' (Additional Settings), includes day/night temperature settings (21 C at 6 H, 18 C at 19 H), an 'Aktivnost auto temperature' toggle (OFF), and a refresh rate dropdown set to '20 S' with a help icon.

Slika 18: Prikaz početne forme internet aplikacije.

Izvor: autor



Kod za komunikaciju s Arduino sustavom i web serverom prikazan je na slici 19. Zadaća prikazanog koda je da prilikom komunikacije između Arduino uređaja i web servera PHP skripta upiše vrijednosti senzora u bazu podataka, uzme postavke i vrati postavke Arduino uređaju. Korištena metoda kojom se poziva PHP skripta je (engl. *Web Request*).

```
<?php
if(isset($_POST["arduino_request"])){
    $arrArduino=explode(";",$_POST["arduino_request"]);
    if(count($arrArduino)>5){
        $temperatura=$arrArduino[0]; $vlaga=$arrArduino[1];
        $pokret=$arrArduino[2]; $voda=$arrArduino[3]; $dim=$arrArduino[4];
        $datum=date("Y-m-d H:i:s");
        $arrDB=$database->Select("SELECT * FROM stanja ORDER BY id DESC LIMIT 1");
        $arrPostavke=$database->Select("SELECT * FROM postavke LIMIT 1");
        echo "$".$arrPostavke[0]["temperatura"].
        "$".$arrPostavke[0]["relay2"].
        "$".$arrPostavke[0]["relay3"].
        "$".$arrPostavke[0]["relay4"].
        "$".$arrPostavke[0]["alarm_aktivnost"].
        "$".$arrPostavke[0]["upali_sirenu"].
        "$".$arrPostavke[0]["osvjezavanje"];
        $spremi=true;
        if(count($arrDB)>0){
            if($arrDB[0]["temperatura"]==temperatura
            && $arrDB[0]["vlaga"]==vlaga
            && $arrDB[0]["dim"]==dim
            && $arrDB[0]["voda"]==voda
            && $arrDB[0]["pokret"]==pokret)
            {$spremi=false;}}
        if($spremi){
            $database->Execute("INSERT INTO stanja (id_korisnik, temperatura,
            vlaga, dim, voda,pokret, datum) VALUES ('.
            ''.SqlInjection('0').'','.
            ''.SqlInjection($temperatura).'','.
            ''.SqlInjection($vlaga).'','.
            ''.SqlInjection($dim).'','.
            ''.SqlInjection($voda).'','.
            ''.SqlInjection($pokret).'','.
            ''.SqlInjection($datum).'''.
            ');");
            $arrST=$database->Select("SELECT * FROM postavke");
        }else
            $database->Execute("UPDATE stanja SET
            datum=''.SqlInjection($datum).'' WHERE id='".$arrDB[0]["id"]."'");
    }
    else{echo "Krivi podatak u stringu";}
}
else{echo "Krivi podatak";}
?>
```

Slika 19: Prikaz PHP skripte pod nazivom kontroler.php.

Izvor: autor

Za komunikaciju između Android uređaja i Web servera također se koristi PHP skripta (prikazana na slici 20). Komunikacija radi tako da Android telefon pošalje zahtjev PHP skripti smještenoj na serveru, a skripta vraća zadnje stanje svih senzora.

```
<?php
include_once("db/mysql.php");
$databse = new MySQL;
header('Content-type: text/plain; charset=utf-8');
if(isset($_POST["device_request"])){
    $arrArduino=explode("¤",$_POST["device_request"]);
    if(count($arrArduino)>0)
    {
        $arrV=$databse->select("SELECT *,
                                (select alarm_aktivnost from postavke) as alarm_aktivnost
                                FROM stanja
                                ORDER BY id DESC LIMIT 1");
        echo $arrV[0]["dim"]."¤".$arrV[0]["voda"]."¤"
            . $arrV[0]["pokret"]."¤".$arrV[0]["alarm_aktivnost"];
    }
    else
        echo "Krivi podatak u stringu";
}
else
    echo "Krivi podatak";
?>
```

Slika 20: Prikaz PHP skripte koja se koristi za komunikaciju s Android uređajem.

Izvor: autor

## 6. Cijena sustava

Za sustav su potrebne domena i server koje treba plaćati svaku godinu. Postoje i besplatni serveri s domenom, ali takvi serveri prikazuju reklame na stranicama. Uz sve spomenuto, potrebna je i internetska veza u objektu na koju će se spojiti Arduino, odnosno *Ethernet Shield*. Cijene koje su navedene u tablici su približne cijene s internetskih stranica <http://www.ebay.com/> i <https://www.avalon.hr/>. Cijene su preračunate iz dolara u kune prema srednjem tečaju Hrvatske narodne banke na dan 06.02.2015, 1USD = 6,542586 kn. Detaljnije cijene prikazane su u tablici 3.

Tablica 3: Približne cijene komponenata koje su korištene u projektu.

<i>Stavka</i>	<i>Približna cijena</i>
Arduino UNO	50,00 kn
Arduino Ethernet Shield	50,00 kn
Releji 4 izlaza	30,00 kn
Adapter za napajanje DC 9V	15,00 kn
Senzor za pokret bežični	80,00 kn
Wireless prijemnik 443MHZ	20,00 kn
Senzor vatre/svijetla	7,00 kn
Senzor vode	12,00 kn
Senzor temperature	24,00 kn
Server i domena godišnje (jeftinija opcija)	150,00 kn
<b>UKUPNO</b>	<b>438,00 kn</b>

Izvor : <http://www.ebay.com/>, <https://www.avalon.hr/>.

## 7. Zaključak

Arduino platforma je široko primjenjiv mikrokontroler brojnih mogućnosti s posebnim naglaskom na jednostavnost njegova korištenja i bogatu programsku podršku za realizaciju različitih projektnih rješenja. Njegova primjena u ovom projektu omogućila je realizaciju za izradu sistema „pametna kuća“. Sustavom, opisanim u ovom projektu, omogućena je komunikacija između različitih mikrokontrolera i internetskih servera s mogućnošću dvosmjerne komunikacije. Sustav nudi mogućnost interaktivnoga upravljanja preko interneta i kao takav bi trebao pronaći svoje mjesto na tržištu u okolnostima sve veće potražnje za poluautomatskim ili potpuno automatskim sustavima. Unutar ovoga rada predviđena je i mogućnost budućega razvoja implementacijom raznih drugih senzora, kamera i mikrokontrolera. Tako cjelokupan sustav nudi različita proširenja i dodatne mogućnosti. Za nesmetani rad sustava potrebna je stalna veza s internetom što nedvojbeno predstavlja problem sustava. Sljedeći nedostatak je potreba zakupa servera na koji će se spremati podatci. Server bi se mogao realizirati s Arduino Ethernet Shieldom, ali problem je što operateri stalno izmjenjuju internet IP adresu te iz tog razloga Arduino uređaj ili internetski preglednik ne zna na koju domenu, odnosno na koju se IP adresu spojiti.

## 8. Literatura

- [1] Mikrokontroler, <http://sh.wikipedia.org/wiki/Mikrokontroler> (06.02.2015.)
- [2] Arduino uređaj, <http://arduino.cc/> (06.02.2015.)
- [3] Android povijest, <http://www.google.com/about/company/history/> (16.04.2015.)
- [4] Android, <https://www.Android.com/> (16.04.2015.)
- [5] Android SDK, <https://developer.Android.com/sdk/index.html/> (16.04.2015.)
- [6] Eclipse, <https://eclipse.org/org/> (16.04.2015.)
- [7] Eclipse, <https://developer.Android.com/tools/debugging/debugging-studio.html>  
(16.04.2015.)
- [8] Arduino Ethernet Shield, <http://developer.Android.com/tools/help/adb.html>  
(06.02.2015.)
- [9] Programski jezik PHP, <http://php.net/> (06.02.2015.)
- [10] Programski jezik PHP, <http://hr.wikipedia.org/wiki/PHP> (06.02.2015.)
- [11] MySQL sustav baze podataka, <https://www.mysql.com/why-mysql/> (06.02.2015.)
- [12] MySQL sustav baze podataka, <http://php.com.hr/66> (06.02.2015.)
- [13] Arduino uređaji:  
[https://en.wikipedia.org/wiki/List\\_of\\_Arduino\\_boards\\_and\\_compatible\\_systems](https://en.wikipedia.org/wiki/List_of_Arduino_boards_and_compatible_systems)
- [14] Arduino Ethernet Shield, <http://arduino.cc/en/Main/EthernetShield> (06.02.2015.)
- [15] Senzor za temperaturu DHT11, <http://www.adafruit.com/product/386> (06.02.2015.)
- [16] Senzor za vodu, <http://www.instructables.com/id/Water-Level-Sensor-Module-for-Arduino-AVR-ARM-STM3/> (06.02.2015.)
- [17] Releji 5V, <http://www.electronics.com/store/1-channel-5v-relay-module-arduino-compatible-bk008-p-259.html> (06.02.2015.)
- [18] Senzor pokreta bežičan, <http://forum.arduino.cc/index.php?topic=208516.0>  
(06.02.2015.)
- [19] Senzor za vatru, [http://robotbase.en.alibaba.com/product/366467533-211851682/Flame\\_Sensor\\_Arduino\\_Compatible.html](http://robotbase.en.alibaba.com/product/366467533-211851682/Flame_Sensor_Arduino_Compatible.html) (06.02.2015.)
- [20] Domena i server, <https://www.avalon.hr/> (06.02.2015.)
- [21] Arduino uređaj,  
<http://bs.scribd.com/doc/212625108/UpravljanjeKoracnimMotorimaPutemArduinoPlatforme#scribd>
- [22] Programski jezik Android, <https://www.Android.com/> (20.03.2015.)

[23] Java Script, <http://www.w3schools.com/js/> (06.04.2015.)

[24] Knjižica "rc-switch", <https://code.google.com/p/rc-switch/> (16.04.2015.)